Fundamenta Informaticae XX (2007) 1–21
IOS Press

On communicating automata with bounded channels

Blaise Genest*, **Dietrich Kuske**†, **and Anca Muscholl**‡

Abstract. We review the characterization of communicating finite-state machines whose behaviors have universally or existentially bounded channels. These results rely on the theory of Mazurkiewicz traces. We investigate the question whether channel bound conditions are decidable for a given communicating finite-state machine.

1. Introduction

Communicating finite-state machines (CFM for short), or equivalently, FIFO channel systems or message passing automata, are a fundamental model for concurrent systems, in which agents cooperate via asynchronous message passing using unbounded buffers. Compared with other models of true concurrency, like Petri nets for instance, these machines are computationally much harder, actually Turing equivalent [9]. Channel systems are the basic model of the standard ITU notation SDL (norm Z.100), and they are widely used in the design of communication protocols. Basic questions arising in formal verification, such as the reachability problem, are undecidable for CFMs (in contrast, reachability is a famous problem in Petri nets, shown to be decidable in [26, 19]).

Motivated by formal verification questions, an important line of research was devoted to identifying variants of CFMs, or approximated behaviors thereof, that are amenable to algorithmic methods. One such example are lossy FIFO systems, which assume that channels are unreliable. On this model, the reachability problem was shown to be decidable [1, 13], albeit of non-primitive recursive complexity [29]. This high complexity is not the primary reason to consider lossy FIFO systems unsatisfactory: First, the assumption that any message can be lost, is rather artificial in practice (a more realistic assumption is that message loss is ruled by probabilities, [30, 4]). Second, more advanced questions like recurrent reachability (including model-checking of liveness properties) are again undecidable in this model [2].

Another approach to obtain decidability of various model checking questions on CFMs is based on the representation of the set of reachable configurations (including the channel contents) by finite automata, see e.g. [5, 6, 7, 10]. Often this approach requires to relax the operations on channels, which yields an over-approximation of the result.

*IRISA/CNRS, Campus de Beaulieu, Rennes, France. Work supported by the ANR projects DocFlow and DOTS.
†Institut für Informatik, Universität Leipzig, Germany. Supported by DAAD-PROCOPE D/0333596.
‡LaBRI, Université Bordeaux, France. Work supported by the ANR projects DocFlow and DOTS.

This paper provides a survey of recent research on subclasses of CFMs that have been shown to be robust w.r.t. two objectives, namely decidability of model-checking problems and good expressive power. The approach taken here goes beyond regular representations of reachable configurations. We use partial order methods for describing the behaviors and for identifying robust subclasses of reliable channel systems. Formally, the behaviors are described by *Message sequence charts* (MSC for short), another ITU standard (Z.120 [18]). The advantage of reasoning about behaviors of CFMs using MSCs is both succinctness and comprehension, since a single diagram subsumes a set of sequential runs of the CFM. The yardstick for expressive power used in this paper is monadic second order logic (MSO for short) over partial orders of MSCs.

An early line of work considered universally bounded MSCs, only. In terms of a CFM, this amounts to saying that every run can be executed with channels of fixed size, no matter how events are scheduled. Equivalently, there exists some (uniform) bound on the number of transitory messages, at any time. Since the size of the communication channels is fixed uniformly, this constraint turns a CFM into a finite state device. Checking that a CFM is universally bounded is undecidable, and some heuristics were proposed for solving this problem [22]. On the positive side, over universally bounded MSCs, the rich theory of regular languages extends very well: automata (CFMs), logic (monadic second order) and MSC-expressions (regular MSC-graphs) are all equivalent [17] (see also [20], which extends the characterization to infinite MSCs). Moreover, model checking in the realm of universally bounded MSC models is decidable, with elementary complexity [3, 28].

The drawback of models with universally bounded communication channels is the limited expressive power. Intuitively, universal channel bounds require message acknowledgments, which can be difficult to impose in general. For instance, basic protocols of producer-consumer type (such as e.g. the USB protocol) are not universally bounded, since the communication is one-way. Therefore, a relaxation of this restriction on channels was proposed in [16, 14] . The idea is to require an *existential bound* on channels. This means roughly that every CFM run must have *some* scheduling of events that respects a given channel bound (other schedules might exceed the bound). In other words, runs *can* be executed with bounded channels, provided that we schedule the events fairly. For instance, in a producer-consumer setting, the scheduling alternates between producer and consumer actions. This requirement is perfectly legitimate in practice, since real life protocols must be executable with limited communication channels. When a channel overflow happens, then the sender stops temporarily until some message is consumed from the queue. For channel systems with existential bounds, the fundamental Kleene-Büchi equivalence of automata, logics and regular expressions was shown to hold in [14]. Regarding model-checking, the complexity remains the same as in the case of universal bounds, [16, 14].

This survey paper is focused on the issue of expressive power for CFMs with universal and existential channel bounds, respectively. We emphasize on the tight relationship that exists between CFMs with channel bounds and *Mazurkiewicz traces* – a concurrent model introduced by A. Mazurkiewicz in the late seventies, for describing the semantics of safe Petri nets. The rich theory of Mazurkiewicz traces (see [11] for a survey) provides a powerful tool when reasoning about the behaviors of CFMs. We survey the results obtained on the expressive power in [17, 20, 14]. In addition, we show that in the Büchi-like characterization obtained for CFMs with existential bounds, the non determinism of CFMs is unavoidable. Moreover, we consider the problem of testing whether a CFM is existentially, or universally bounded, respectively. We show roughly that the only case where this problem has a solution is when we assume that the channel bound is known and the CFM is deadlock-free. Both these assumptions are motivated by applications, since concrete systems use bounded memory and communication protocols are in general deadlock-free.

Overview. In Section 2 we define communicating finite-state machines whose behavior is investigated in this paper. The following Section 3 introduces formalisms used in this investigation – Mazurkiewicz traces, message sequence charts and monadic second order logic (MSO). In this section we also introduce a normal form for MSCs, that corresponds to an optimal linear execution w.r.t. channel bounds. Section 4 deals with universally bounded CFMs. In this setting, it presents the known equivalence between CFMs, MSO, and regular sets of traces. As new results, we consider the problem of deciding whether a deadlock-free CFM is universally bounded or not. The following Section 5 elaborates these techniques and results further in the setting of existentially bounded CFMs. In particular, we show that deterministic CFMs are not sufficient in this case.

Related work. Existential channel bounds appear in [21] and implicitly in [15] (called there realizable HCMSCs). The expressive power and model checking issues for universally bounded channels are considered in [3, 28, 17, 20, 23]. Without the restriction of universally bounded channels, [24, 25] shows how to use representative executions in model checking against MSO properties and [16] does this against MSC-graph properties. As in [17, 25, 14] we use here the logic that talks about the *partial order* of an MSC. The paper [8] shows that the existential fragment of the weaker MSO logic based on the immediate successor is expressively equivalent to CFMs without any restrictions.

2. Definitions

The communication framework used in our paper is based on sequential processes that exchange asynchronously messages over point-to-point, error-free FIFO channels. Let $\mathcal{P}$ be a finite set of process identities that we fix throughout this paper. Furthermore, let $\mathrm{Ch} = \{(p, q) \in \mathcal{P}^2 \mid p \neq q\}$ denote the set of *channels*. Processes act by either sending a message, that is denoted by $p!q$ meaning that process p sends to process q, or by receiving a message, that is denoted by $p?q$, meaning that process p receives from process q. For any process $p \in \mathcal{P}$, we define a local alphabet (set of event types on p) $\Sigma_p = \{p!q, p?q \mid q \in \mathcal{P} \setminus \{p\}\}$ and set $\Sigma = \bigcup_{p\in\mathcal{P}} \Sigma_p$. For the rest of the paper, whenever a pair of processes $p, q \in \mathcal{P}$ communicates, we will implicitly assume that $p \neq q$, i.e., $(p, q) \in \mathrm{Ch}$.

The most natural formalism to describe (asynchronous) communication protocols are *communicating finite-state machines* (CFM for short) [9]. CFMs are a basic model for distributed algorithms based on asynchronous message passing between concurrent processes:

Definition 2.1. A *communicating finite-state machine* (*CFM*) is a tuple $\mathcal{A} = (C, (\mathcal{A}_p)_{p\in\mathcal{P}}, F)$ where

- C is a finite set of *message contents* or *control messages*.
- $\mathcal{A}_p = (S_p, \rightarrow_p, \iota_p)$ is a finite labeled transition system over the alphabet $\Sigma_p \times C$ for any $p \in \mathcal{P}$ (i.e., $\rightarrow_p \subseteq S_p \times (\Sigma_p \times C) \times S_p$) with initial state $\iota_p \in S_p$.
- $F \subseteq \prod_{p\in\mathcal{P}} S_p$ is a set of global final states.

The CFM $\mathcal{A}$ is *deterministic* [17] if

- $s \xrightarrow{p!q,m_1}_p s_1$ and $s \xrightarrow{p!q,m_2}_p s_2$ implies $s_1 = s_2$ and $m_1 = m_2$
- $s \xrightarrow{p?q,m}_p s_1$ and $s \xrightarrow{p?q,m}_p s_2$ implies $s_1 = s_2$.

The notion of determinism used here originates from [17]. For instance, it can be justified in the setting of distributed supervision, where some distributed plant is extended with a distributed automaton that can attach additional message contents to messages that are exchanged by components of the plant. Thus, the controlling automaton has control over its next state as well as over the message content it attaches to some message. But it does not have control as to whether the next action is sending to or receiving from some particular channel. If the plant decides to execute a receive event $p?q$, then the controlling automaton can only receive the first message of the channel, i.e., should be prepared to receive distinct messages.

In order to describe the behavior of a CFM, one can transform it naturally into a sequential, potentially infinite transition system whose states consist of a $\mathcal{P}$-tuple of local states as well as the contents of the channels. More precisely, one defines from the CFM $\mathcal{A} = (C, (\mathcal{A}_p)_{p\in\mathcal{P}}, F)$ the $(\Sigma \times C)$-labeled, infinite transition system $T_\mathcal{A}$ as follows. A state of $T_\mathcal{A}$ consists of a $\mathcal{P}$-tuple of local states and of channel contents of $\mathcal{A}$, i.e., it is an element $((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}})$ of $\prod_{p\in\mathcal{P}} S_p \times \prod_{(p,q)\in\mathrm{Ch}} C^*$. For two states, an action $a \in \Sigma_p$, and a control message $c \in C$, we have

$$((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}}) \xrightarrow{a,c} ((s'_p)_{p\in\mathcal{P}}, (w'_{p,q})_{(p,q)\in\mathrm{Ch}})$$

if

- $s_p \xrightarrow{a,c}_p s'_p$ is a transition of the local machine $\mathcal{A}_p$ and $s_q = s'_q$ for $q \neq p$.
- Send events: if $a = p!q$, then $w'_{p,q} = w_{p,q}c$ (i.e., message c is inserted into the channel from p to q) and $w_{p',q'} = w'_{p',q'}$ for $(p', q') \neq (p, q)$ (i.e., all other channels are unchanged)
- Receive events: if $a = p?q$, then $w_{q,p} = cw'_{q,p}$ (i.e., message c is deleted from the channel from q to p) and $w_{q',p'} = w'_{q',p'}$ for $(q', p') \neq (q, p)$ (i.e., all other channels are unchanged).

A run of $T_\mathcal{A}$ is as usual a sequence $d_1, (a_1, c_1), d_2, (a_2, c_2), \ldots, (a_n, c_n), d_{n+1}$ with d_i states of $T_\mathcal{A}$, $a_i \in \Sigma$ and $c_i \in C$ such that $d_i \xrightarrow{a_i,c_i} d_{i+1}$ for all suitable i. It is accepting if $d_1 = ((\iota_p)_{p\in\mathcal{P}}, (\varepsilon)_{(p,q)\in\mathrm{Ch}})$ and $d_{n+1} = (f, (\varepsilon)_{(p,q)\in\mathrm{Ch}})$ for some $f \in F$. Finally, we define $L(T_\mathcal{A}) \subseteq \Sigma^*$ by projecting the control messages and states out of accepting runs: it is the set of words $a_1a_2\cdots a_n$ such that there exists an accepting run $d_1, (a_1, c_1), d_2, (a_2, c_2), \ldots, (a_n, c_n), d_{n+1}$.

A CFM is called *deadlock-free*, if $F = \prod_{p\in\mathcal{P}} S_p$ and from every reachable state of $T_\mathcal{A}$ we can reach a state where all channels are empty.

3. Partial orders of Mazurkiewicz traces and CFMs

We consider in this section two different kinds of partial orders, Mazurkiewicz traces and runs of CFMs. Then we establish a relationship between these partial orders, which is the basis of several results of expressiveness and decidability for subfamilies of CFMs.

3.1. Partial orders

Mazurkiewicz traces [27] have been introduced in computer science for describing the behavior of safe Petri nets. Their essence is to describe the semantics of a concurrent system by a (static) relation of independence between actions. Formally, a *trace alphabet* is a pair (Ω, I) consisting of an alphabet Ω and a symmetric and irreflexive relation $I \subseteq \Omega^2$. The relation I will be referred to as the *independence relation*; its complement $D = \Omega^2 \setminus I$ is the *dependence relation*.

A *Mazurkiewicz trace* is an Ω-labeled partial order $(E, \leq, \lambda)$ (up to isomorphism), with the labeling $\lambda : E \to \Omega$ satisfying the following conditions, for any events $e, f \in E$:

- if e is an immediate predecessor of f (denoted as $e \lessdot f$), then $(\lambda(e), \lambda(f)) \in D$, and
- if e and f are incomparable, then $(\lambda(e), \lambda(f)) \in I$.

Partial orders also arise naturally when we describe runs of CFMs. Instead of viewing the CFM as an infinite transition system, we can visualize the runs by means of diagrams called *message sequence charts* (MSC for short).

We define message sequence charts as Σ-labeled posets $(E, \leq, \lambda)$, and we write $P(e)$ for the process on which event e is located. That is, we let $P(e) = p$ if $\lambda(e) \in \Sigma_p$. In addition, we define two relations $\leq_P$ and $<_m$ on events:

- $e \leq_P f$ iff $P(e) = P(f)$ and $e \leq f$.
- $e <_m f$ iff $\lambda(e) = p!q$, $\lambda(f) = q?p$, and $|\{e' \mid \lambda(e') = p!q, e' \leq e\}| = |\{f' \mid \lambda(e') = q?p, f' \leq f\}|$, for some $p, q \in \mathcal{P}$.

The idea is that $\leq_P$ describes the order of the events executed by the sequential processes. If $P(e) = P(f) = p$ and $e < f$, we also write $e <_p f$. Moreover, if there is no event g with $P(g) = p$ and $e < g < f$, then we write $e \lessdot_p f$. The relation $<_m$ describes the matching send and receive events, under the assumption that message channels are FIFO.

Definition 3.1. A *message sequence chart* is a Σ-labeled poset $M = (E, \leq, \lambda)$ (up to isomorphism) satisfying

- $\leq = (\leq_P \cup <_m)^*$,
- $P^{-1}(p) \subseteq E$ is linearly ordered for any $p \in \mathcal{P}$, and
- $|\lambda^{-1}(p!q)| = |\lambda^{-1}(q?p)|$ for any $(p, q) \in \mathrm{Ch}$.

An example MSC is shown in Figure 2. If we replace the last item of the definition above by $|\lambda^{-1}(p!q)| \geq |\lambda^{-1}(q?p)|$ for any $(p, q) \in \mathrm{Ch}$, then we speak about *prefix MSC*.

Any linear extension of a labeled partial order $(E, \leq, \lambda)$ is called a *linearization* of it. We represent it as a word $u = u_1 \cdots u_n$ over the alphabet Σ, if $\lambda : E \to \Sigma$. Thus, the set $\mathrm{Lin}(M)$ of linearizations of the MSC M is a subset of Σ^*, and the set of linearizations $\mathrm{Lin}(t)$ of a trace t is a subset of Ω^*. For a set (or language) of partial orders $\mathcal{M}$, we write $\mathrm{Lin}(\mathcal{M}) = \bigcup_{M \in \mathcal{M}} \mathrm{Lin}(M)$. For any $w \in \Sigma^*$, $a \in \Sigma$, we denote as usual by $|w|_a$ the number of occurrences of a in w.

For MSCs, the relation between the partial order and its linearizations is tighter: starting with any word w from Σ^* that satisfies $|v|_{p!q} \geq |v|_{q?p}$ for any prefix v and every channel $(p, q) \in \mathrm{Ch}$, there exists a unique prefix MSC M such that w is a linearization of M. We denote this prefix MSC as $\mathrm{msc}(w)$. If $w \in \Sigma^*$ does not satisfy the above condition on channels, then $\mathrm{msc}(w)$ is undefined.

Runs of CFMs can be also viewed as (prefix) MSCs. Let $\mathcal{A}$ be a CFM, and consider the set of labelings of runs $L(T_{\mathcal{A}}) \subseteq \Sigma^*$. It can be shown easily that for every MSC M with $\mathrm{Lin}(M) \cap L(T_{\mathcal{A}}) \neq \emptyset$ we have $\mathrm{Lin}(M) \subseteq L(T_{\mathcal{A}})$. We denote by $\mathcal{L}(\mathcal{A})$ the language of the CFM $\mathcal{A}$, that is, the set of MSCs associated with accepting runs of $\mathcal{A}$: $\mathcal{L}(\mathcal{A}) = \{\mathrm{msc}(w) \mid w \in L(T_{\mathcal{A}}\}$. By the above remarks, we have $\mathrm{Lin}(\mathcal{L}(\mathcal{A})) = L(T_{\mathcal{A}})$.

Definition 3.2. Let $B > 0$ be an integer. A word (linearization) $w \in \Sigma^*$ is called *B-bounded* if $|v|_{p!q} - |v|_{q?p} \leq B$, for all prefixes v of w and all $(p, q) \in \mathrm{Ch}$. An MSC $M = (E, \leq, \lambda)$ is called *universally B-bounded* if every linearization of M is so. A set of MSCs is universally B-bounded if each of its elements is so. A CFM $\mathcal{A}$ is universally B-bounded if every configuration $((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}})$ of some accepting run of $T_{\mathcal{A}}$ satisfies $|w_{p,q}| \le B$ for all $(p, q) \in \mathrm{Ch}$.

A CFM (set of MSCs, respectively) is called universally bounded if it is universally B-bounded for some $B > 0$.

From the remark above it is easy to see that a CFM $\mathcal{A}$ is universally B-bounded if and only if $\mathcal{L}(\mathcal{A})$ is universally B-bounded.

Let $\mathrm{Lin}_B(M)$ denote the set of B-bounded linearizations of an MSC M, and $\mathrm{Lin}_B(\mathcal{M})$ is to be understood similarly for a set of MSCs $\mathcal{M}$. In any case $\mathrm{Lin}_B(\mathcal{M}) \subseteq \mathrm{Lin}(\mathcal{M})$, with equality if and only if $\mathcal{M}$ is universally B-bounded.

3.2. Traces and MSCs

We describe in this section a tight link between Mazurkiewicz traces and universally B-bounded MSCs, due to [20]. Let (Ω, I) be the trace alphabet with $\Omega = \Sigma \times \{0, \ldots, B - 1\}$. The dependence relation $D \subseteq \Omega \times \Omega$ is given by $(x, i)D(y, j)$ if either $P(x) = P(y)$ or $\{(x, i), (y, j)\} = \{(p!q, n), (q?p, n)\}$ for some p, q, n. Clearly, $I = \Omega^2 \setminus D$ is symmetric and irreflexive, hence (Ω, I) is a trace alphabet.

For an Ω-labeled poset $t = (E, \le, \lambda)$, let $\mathrm{proj}(t)$ denote the Σ-labeled poset $(E, \le, \pi \circ \lambda)$ where $\pi : \Omega \to \Sigma$ is the projection to the first component.

The encoding $\mathrm{tr}(M)$ of an MSC $M = (E, \le, \lambda)$ is obtained by numbering the events of the same type modulo B: $\mathrm{tr}(M) = (E, \le, \lambda')$ such that $\lambda'(e) = (\lambda(e), n)$ with $n = |\{e' < e \mid \lambda(e') = \lambda(e)\}| \bmod B$.

In general, the partial order $\mathrm{tr}(M)$ is no Mazurkiewicz trace. Consider, for instance, the MSC $M = (E, \le, \lambda)$ with linearization $(1!2)(1!2)(2?1)(2?1)$ and $B = 1$. Then $\mathrm{tr}(M) = (E, \le, \lambda')$ with $\lambda'(e) = (\lambda(e), 0)$ for any $e \in E$. Hence, in $\mathrm{tr}(M)$, the first occurrence of $2?1$ and the second of $1!2$ carry dependent labels, but these events are incomparable, i.e., $\mathrm{tr}(M)$ is indeed not a trace.

Lemma 3.1. [20] Let $M = (E, \le, \lambda)$ be a universally B-bounded MSC, then the partial order $\mathrm{tr}(M)$ is a trace over the alphabet (Ω, D) and we have $M = \mathrm{proj}(\mathrm{tr}(M))$.

Note that the converse implication in the above lemma does not hold, in general. Consider the MSC $M = (E, \le, \lambda)$ with (unique) linearization $w = (1!2)(1!2)(1!3)(3?1)\ (3!2)(2?3)(2?1)(2?1)$ and $B = 1$. Then M is not universally 1-bounded, but $\mathrm{tr}(M)$ is a trace, since it is linearly ordered. The reader can verify that an MSC is universally B-bounded if and only if in the partial order of $\mathrm{tr}(M)$, between any two consecutive nodes labeled by $(p!q, n)$ there is a node labeled by $(q?p, n)$.

Lemma 3.1 provides the basis for a quadratic-time algorithm that checks that an MSC is universally B-bounded (see also [21] for an alternative approach). It verifies that the partial order $\mathrm{tr}(M)$ satisfies the two conditions in the definition of Mazurkiewicz traces and that in between any two $(p!q, n)$-labeled nodes, there is a $(q?p, n)$-labeled one.

3.3. Optimal linearizations

We present in this section an algorithm to compute a linearization $\mathrm{OPT}(M) \in \Sigma^*$ of the MSC $M = (E, \le, \lambda)$ that is B-bounded, for the least possible B.

The algorithm computes a linearization $\mathrm{OPT}(M)$ incrementally: If M is empty, then $\mathrm{OPT}(M) = \varepsilon$. Otherwise, suppose that we have already computed the linearization of a prefix of M, with set of events $F \subseteq E$. We choose now the next event $e \in E \setminus F$ among those events such that $F \cup \{e\}$ is downward closed (that is, for every $f \leq g$ with $g \in F \cup \{e\}$, we have $f \in F \cup \{e\}$ as well). Let G denote the set of such candidates. If G contains some receive event e, then we add e to F. Otherwise, we add to F a send event on a channel (p, q) that has the least number $|\{f \in F \mid \lambda(f) = p!q\}| - |\{f \in F \mid \lambda(f) = q?p\}|$ of pending messages in F from p to q. Ties are broken using some fixed linear order $\sqsubseteq$ on the set of channels Ch, we always take the event that involves the largest possible channel.

Proposition 3.1. Let $\mathrm{OPT}(M)$ be the linearization computed by the above algorithm on MSC M. Let also $B \in \mathbb{N}$ be minimal such that $\mathrm{OPT}(M)$ is B-bounded. Then no linearization of M is $(B-1)$-bounded.

Proof:
Let $w \in \mathrm{Lin}(M)$ be some linearization. Let $x \in \Sigma^*$ and $a \in \Sigma$ such that xa is the minimal prefix of $\mathrm{OPT}(M)$ that is not $(B-1)$-bounded. Then there exists $(p, q) \in \mathrm{Ch}$ with $a = p!q$ and $n_{r,s} \leq B - 1 = n_{p,q}$ for any $(r, s) \in \mathrm{Ch}$ where $n_{r,s} = |x|_{r!s} - |x|_{s?r}$. Let zb be the minimal prefix of w such that $\mathrm{msc}(zb)$ is no prefix of $\mathrm{msc}(x)$ (with $z \in \Sigma^*$ and $b \in \Sigma$). Then also xb is a linearization of some prefix of M. Hence, by the choice of a in the algorithm, there exist $(r, s) \in \mathrm{Ch}$ with $b = r!s$ and $n_{r,s} \geq n_{p,q} = B - 1$ (in particular, $n_{r,s} = n_{p,q} = B - 1$). Since $\mathrm{msc}(z)$ is a prefix of $\mathrm{msc}(x)$, we have $|z|_{s?r} \leq |x|_{s?r}$. In addition, $\mathrm{msc}(zb) = \mathrm{msc}(z\, r!s)$ and $\mathrm{msc}(x)$ are prefixes of M; hence $|z|_{r!s} = |x|_{r!s}$. Together, this implies $|z|_{r!s} - |z|_{s?r} \geq |x|_{r!s} - |x|_{s?r} = n_{r,s} = n_{p,q} = B - 1$. Hence zb (and therefore its extension w) is not $(B - 1)$-bounded. □

Since channels are in general unbounded, the set $\{\mathrm{OPT}(M) \mid M \text{ MSC}\}$ cannot be regular. The following proposition shows that this is the only obstacle, i.e., if we restrict to channels of bounded size, then the optimality of a linearization can be tested by an automaton.

Proposition 3.2. Let $B > 0$ be an integer. There exists a polynomial-size automaton $\mathcal{A}$ such that, for any MSC M and any $u \in \mathrm{Lin}_B(M)$, we have $u \in L(\mathcal{A})$ if and only if $u \neq \mathrm{OPT}(M)$.

Proof:
Note that the word $u \in \mathrm{Lin}(M)$ does *not* equal $\mathrm{OPT}(M)$ iff there exist $v, w \in \Sigma^*$ and $a \in \Sigma$ with $u = vaw$ and $p, q, r, s \in \mathcal{P}$ such that $r \neq p$ and (1) or (2) hold

(1) $a = p!q$ and one of the following holds:

- $b = r?s$ is the first action from Σ_r in w, and $|v|_{r!s} > |v|_{s?r}$;
- $b = r!s$ is the first action from Σ_r in w and either $|v|_{p!q} - |v|_{q?p} > |v|_{r!s} - |v|_{s?r}$, or $|v|_{p!q} - |v|_{q?p} = |v|_{r!s} - |v|_{s?r}$ and $(p, q) \sqsubset (r, s)$.

(2) $a = p?q$, $(p, q) \sqsubset (r, s)$, $b = r?s$ is the first action from Σ_r in w, and $|v|_{r!s} > |v|_{s?r}$.

The reason is that, in any of these cases, the algorithm would have preferred b over a after v. For instance, for $a = p!q$ and $b = r!s$, the algorithm has the choice between a and b and prefers b, since the channel (r, s) is either less filled than (p, q), or equally filled and (r, s) has higher priority than (p, q). For $b = r?s$, the condition $|v|_{r!s} > |v|_{s?r}$ ensures that events a and b are simultaneously candidates after v. To check the above conditions, the automaton $\mathcal{A}$ guesses the processes p, q, r, s and keeps track of the values $|v|_{p!q} - |v|_{q?p}$ and $|v|_{r!s} - |v|_{s?r}$. Since we are only interested in B-bounded linearizations, this can be done with $|\mathcal{P}|^4 \cdot (B + 1)^2$ many states. □

3.4. Monadic second order logic

Logic is a classical formalism used to describe properties of various structures, like words, trees, traces, graphs etc. This also applies to structures like MSCs. We consider here monadic second order logic, with the following syntax:

Definition 3.3. For a set $\mathcal{R}$ of binary relations, *MSO*$(\mathcal{R})$*-formulas over the alphabet* Γ are defined by the syntax

$$\varphi ::= a(x) \mid R(x,y) \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists X \varphi \mid \exists x \varphi$$

where $R \in \mathcal{R}$, $a \in \Gamma$, x, y are first order variables, and X is a second order variable.

An MSO($\leq$)-formula over an alphabet Γ can be interpreted on Γ-labeled partial orders $M = (E, \leq, \lambda)$ with $\lambda : E \to \Gamma$ as usual, by letting $M \models a(x)$ if $\lambda(x) = a$ and $M \models x \leq y$ if $x \leq y$. Further relations in $\mathcal{R}$ used here are the message order $<_m$, the process order $(<_p)_{p\in\mathcal{P}}$, and the immediate process successor $(\lessdot_p)_{p\in\mathcal{P}}$. Such an MSO($\mathcal{R}$)-formula over the alphabet Σ can then be interpreted on an MSC $M = (E, \leq, \lambda)$ as expected.

For an MSO($\mathcal{R}$)-formula φ over Σ without free variables, let $\mathcal{L}(\varphi)$ denote the set of MSCs that satisfy φ. We also consider existential monadic second order logic (EMSO). An EMSO formula is of the form $\exists X_1 \ldots X_n \varphi$ with φ a first order formula, i.e., without second-order quantification

We discuss now some differences arising from the use of different predicates from $\mathcal{R}$. First, the full logics MSO($\leq, <_m$) and MSO($(\lessdot_p)_{p\in\mathcal{P}}, <_m$) are equally expressive, but the existential fragment of the former could be more expressive than the existential fragment of the latter (which is the logic considered in [8]). From [8] (Cor. 5.7) we know that MSO($\leq$) and EMSO($(\lessdot_p)_{p\in\mathcal{P}}, <_m$) are incomparable. Furthermore, we will show later that (universally and existentially) B-bounded sets of MSCs behave better, since they provide the equivalence between MSO($\leq$), EMSO($\leq$) and EMSO($(\lessdot_p)_{p\in\mathcal{P}}, <_m$).

4. The behavior of universally bounded CFMs

This section is devoted to universally bounded CFMs and MSCs. First we recall the Büchi-like characterization of universal boundedness in terms of CFMs, MSCs and logics. Then we present some (un)decidability results related to universal channel bounds.

4.1. Büchi characterization of universally bounded behaviors

Let (Ω, I) be a trace alphabet. A set L of traces over (Ω, I) is *regular* if its set of linearizations $\mathrm{Lin}(L) \subseteq \Omega^*$ is regular.

Lemma 4.1. Let $\mathcal{M}$ be a set of MSCs. If $\mathrm{Lin}(\mathcal{M})$ is regular, then there exists a regular language of traces L over (Ω, I) such that $\mathcal{M} = \mathrm{proj}(L)$.

Proof:
Since $\mathrm{Lin}(\mathcal{M})$ is regular, there is some $B > 0$ such that any linearization in $\mathrm{Lin}(\mathcal{M})$ is B-bounded. In particular, any $M \in \mathcal{M}$ is universally B-bounded. Set (Ω, I) and the mappings $\mathrm{tr}, \mathrm{proj}$ as in Section 3. Since any $M \in \mathcal{M}$ is universally B-bounded, the Ω-labeled poset $\mathrm{tr}(M)$ is a trace over (Ω, I) (Lemma 3.1). Let $K = \{\mathrm{tr}(M) \mid M$ is universally B-bounded$\}$ and $L = \{\mathrm{tr}(M) \mid M \in \mathcal{M}\}$. Then, certainly, $\mathcal{M} = \mathrm{proj}(L)$ and it remains to show that L is regular. For this, note that a word

from $\mathrm{Lin}(K)$ is a linearization of some trace $\mathrm{tr}(M)$ in L iff its projection via proj is a linearization of $\mathrm{proj}(\mathrm{tr}(M)) = M$ (the right to left implication follows from the fact that the partial orders of M and $\mathrm{tr}(M)$ are isomorphic, cf. Lemma 3.1). By assumption, $\mathrm{Lin}(\mathcal{M})$ is regular, and [20, Lemmas 3.6, 3.7] shows that $\mathrm{Lin}(K)$ is regular. Hence L is regular, too. $\square$

The next theorem provides the characterization of universally bounded CFMs (with given channel bound) in terms of monadic second-order logic and of regular linearizations. For lack of space, we have omitted a third characterization, in terms of regular CMSC-graphs [15], that corresponds to a kind of regular expressions of communication events. The results given below were obtained in [17], and [20] extended them to sets of infinite MSCs (and CFMs with Muller acceptance). The most difficult part of the theorem is the construction of a deterministic CFM from a regular set $\mathrm{Lin}(\mathcal{M})$, since it amounts to give an algorithm of distributed synthesis. The original approach of [17] consists in adapting Zielonka's construction of deterministic asynchronous automata [32] for regular trace languages to the setting of universally B-bounded MSCs. Later, [20] made the connection between MSCs and traces explicit (see Section 3) and gave a simplified construction of deterministic CFMs, that uses Zielonka's construction as a black-box.

Theorem 4.1. [17, 20] Let B be a positive integer and $\mathcal{M}$ a set of universally B-bounded MSCs. Then the following assertions are equivalent:

1. $\mathrm{Lin}(\mathcal{M})$ is regular.

2. $\mathcal{M}$ is the language of some CFM.

3. $\mathcal{M}$ is the language of some *deterministic* CFM.

4. $\mathcal{M}$ is the language of some MSO($\leq$) formula.

5. $\mathcal{M}$ is the language of some formula of EMSO($(\lessdot_p)_{p\in\mathcal{P}}, <_m$), EMSO($\leq$), or EMSO($\leq, <_m$), respectively.

Let us state a few ideas involved in the proof of the above theorem. It is easy to see that any universally bounded CFM $\mathcal{A}$ has a regular set of linearizations. The converse, as mentioned above, can be shown using Lemma 4.1 and Zielonka's construction. The main idea is to simulate the execution of a deterministic asynchronous automaton $\mathcal{A}$ on $\mathrm{tr}(M)$ by a deterministic CFM $\mathcal{B}$ on M. Since the partial orders of M and $\mathrm{tr}(M)$ are isomorphic, the necessary information about local states of $\mathcal{A}$ that are visible for an event on $\mathrm{tr}(M)$ is also available on M, by storing it in the local states of $\mathcal{B}$. As for the logic part, the equivalence between MSO($\leq$) and the regular set of linearizations follows without much difficulty from Lemma 4.1 together with [31, 12], that shows a similar result for traces. Finally, the last item in the theorem is obtained with the usual simulation of automata by EMSO. We note that for universally B-bounded MSCs, the message relation $<_m$ can be expressed in terms of the partial order $\leq$, hence we obtain EMSO($\leq, <_m$) $=$ EMSO($\leq$). The idea is that the trace encoding of Section 3.2 corresponds to additional existentially quantified set variables, one for each set of events with trace label (σ, n). This allows to say that the receive matching a $(p!q, n)$-send e is the first node f after e, with label $(q?p, n)$. In addition, we need to ensure by a formula of EMSO($\leq$) that the model is a universally B-bounded MSC. But this is easy, see remarks after Lemma 3.1.

4.2. Testing universal bounds

In this section we show that in general, the property of universal boundedness is hard to check. It is not very surprising that one cannot check whether an *arbitrary* CFM is universally bounded, since CFMs are Turing-equivalent devices. We strengthen this observation by showing that undecidability holds even assuming that the CFM is deterministic and deadlock-free. If we provide the bound B as input, the problem of testing whether a deterministic CFM is universally B-bounded is still undecidable. However, for *deadlock-free* CFMs we obtain decidability.

For the undecidability results, we use the following encoding of a deterministic Turing machine TM by a deterministic and deadlock-free CFM. We will define the CFM $\mathcal{A}_{\mathrm{TM}}$ on two processes $1, 2$. A configuration of TM will be encoded as a sequence of messages with contents $m_1, \ldots, m_{k-1}, q, m_k, \ldots, m_n$, meaning that TM is in state q, the tape contents is $m_1 \cdots m_n$ and the head position is k. With this encoding, it suffices to know three consecutive messages of this sequence in order to compute deterministically the i-th symbol m'_i of the next configuration.

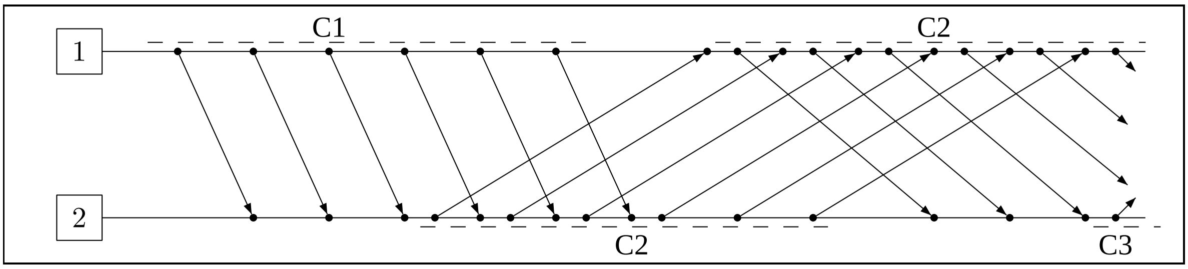

Figure 1. Encoding a Turing machine by a deadlock-free deterministic CFM.

The CFM works as follows. First, process 1 sends the initial configuration C_1 to process 2. Then it resends any configuration C_i received from process 2 back to process 2, interleaving receives with sends. Process 2 receives a configuration C_i from process 1 and sends the successor configuration C_{i+1} to process 1, also interleaving receives with sends.

In order to obtain a deterministic CFM, process 2 awaits the first three symbols from process 1 before it actually starts sending the next configuration, one send for each receive. Then, it finishes by sending three messages (or two, or four, depending on the length of C_{i+1}) that end the successor configuration C_{i+1}.

More formally, we denote by $w = a_1 \cdots a_n \parallel_k b_1 \cdots b_m$ the k-delayed fair shuffle of $a_1 \cdots a_n$ and $b_1 \cdots b_m$, defined as $w = a_1 a_2 \cdots a_k b_1 a_{k+1} b_2 \cdots a_n b_{n-k+1} \cdots b_m$. The language of events on process 1 is $\mathrm{SC}_1 \prod_{i \geq 2}(\mathrm{RC}_i \parallel_1 \mathrm{SC}_i)$, where SC_i means sending configuration C_i to process 2 and RC_i means receiving C_i from process 2. Similarly, the language of process 2 is $\prod_{i \geq 1}(\mathrm{RC}_i \parallel_3 \mathrm{SC}_{i+1})$.

Proposition 4.1. Let $B > 0$. It is undecidable whether a deterministic CFM is universally B-bounded.

Proof:
Using the above encoding, we reduce the halting problem on empty input for deterministic Turing machines to the test of the universal B-boundedness of a CFM $\mathcal{A}$. So let TM be some deterministic Turing machine and let $\mathcal{A}_{\mathrm{TM}} = (C, (\mathcal{A}_p)_{p \in \mathcal{P}}, F)$ be the deterministic CFM constructed above, with F corresponding to halting configurations of TM (and where process 1 stops resending the current configuration).

Let $q, r \notin \mathcal{P}$ be two new processes. We define now the CFM $\mathcal{B} = (C, (\mathcal{A}_p)_{p \in \mathcal{P} \cup \{q,r\}}, F')$ with:

- $\mathcal{A}_q = (\{s_0, \ldots, s_{B+1}\}, \rightarrow_q, s_0)$ with $s_i \xrightarrow{q!r,c}_q s_{i+1}$ for all $i \leq B$, and $s_{B+1} \xrightarrow{q?r,c}_q s_{B+1}$.
- $\mathcal{A}_r = (\{t_0, \ldots, t_{B+1}\}, \rightarrow_r, t_0)$ with $t_i \xrightarrow{r!q,c}_r t_{i+1}$ for all $i \leq B\}$, and $t_{B+1} \xrightarrow{r?q,c}_r t_{B+1}$.
- $F' = F \times \{(s_{B+1}, t_{B+1})\}$

where c is some fixed control message from C.

Actually, the CFM $\mathcal{B}$ simply adds to $\mathcal{A}_{\text{TM}}$ a behavior on $\{q, r\}$ that consists in $B + 1$ messages from q to r, that cross $B + 1$ messages from r to q. This MSC M_B is not universally B-bounded. Hence $\mathcal{L}(\mathcal{B})$ is obtained by simply adjoining M_B to any MSC from $\mathcal{L}(\mathcal{A}_{\text{TM}})$. Hence either $\mathcal{L}(\mathcal{A}) = \emptyset = \mathcal{L}(\mathcal{B})$, implying that $\mathcal{B}$ is universally B-bounded. Or $\mathcal{L}(\mathcal{A}) \neq \emptyset$ and $\mathcal{B}$ is not universally B-bounded. □

Proposition 4.2. It is undecidable whether a deterministic and deadlock-free CFM is universally bounded.

Proof:
Let TM be a deterministic Turing machine. The existence of some $B > 0$ such that every configuration of TM reached from the empty input is of size at most B is undecidable (for otherwise, we could decide the halting problem of TM). We reduce this undecidable problem to the question whether a deterministic and deadlock-free CFM is universally bounded.

Let $\mathcal{A}_{\text{TM}}$ be the deterministic CFM constructed above. It is easy to check that if every configuration of TM is of size bounded by B, then the CFM $\mathcal{A}_{\text{TM}}$ is universally B-bounded. Conversely, if a reachable configuration is of size greater than B, then its associated sends (without the matching receives) will require a channel size larger than B. Hence, $\mathcal{A}_{\text{TM}}$ is universally bounded iff TM is bounded.

We obtain that the CFM $\mathcal{A}_{\text{TM}}$ is deadlock-free by defining all states as final, together with the following modification: after sending a complete configuration C_i, process 1 can stop forwarding messages to process 2, it will only receive RC_{i+1}. Hence from any configuration a final state can be reached, that is, $\mathcal{A}_{\text{TM}}$ is deadlock-free. Notice that the CFM is still deterministic because process 2 has no choice, and the only choices of process 1 are between a receive and a send. □

Remark 4.1. Our definition of universally bounded CFM differs actually from the one used in [17], who requires that all configurations of *any* run of the CFM (not only accepting ones) are B-bounded. Note that for the CFM defined in Proposition 4.2 all states are final, so the result also holds w.r.t. the definition of universal boundedness used by [17]. On the other hand, the question considered in Proposition 4.1 becomes decidable in the setting of [17].

For a language $L \subseteq \Sigma^*$ we denote by $\text{Pref}(L)$ the set of prefixes of L. Similarly, for a CFM $\mathcal{A}$, $\text{Pref}(\mathcal{A}) \subseteq \Sigma^*$ stands for $\text{Pref}(L(T_{\mathcal{A}}))$. Let $B > 0$, then we set $\text{Pref}_B(\mathcal{A})$ as the subset of $\text{Pref}(\mathcal{A})$ consisting of B-bounded words, only. Notice that if an MSC M is universally B-bounded, then any prefix of M is universally B-bounded.

Proposition 4.3. Let $\mathcal{A}$ be a CFM, and $B > 0$. Then $\mathcal{A}$ is universally B-bounded if and only if every word in $\text{Pref}_{B+1}(\mathcal{A})$ is B-bounded.

Proof:
If $\mathcal{A}$ is universally B-bounded, then so is $\mathrm{Pref}(\mathcal{A})$, which shows the first implication.

Conversely, assume that $\mathcal{A}$ is not universally B-bounded, and consider some $w = a_1 \cdots a_n \in L(T_{\mathcal{A}})$ that is not B-bounded. Clearly, if $a_1 \cdots a_i$ is K-bounded, then $a_1 \cdots a_{i+1}$ is $(K+1)$-bounded, for any i, K. Thus there exists some $i \le n$ such that $a_1 \cdots a_i$ is not B-bounded, but belongs to $\mathrm{Pref}_{B+1}(\mathcal{A})$. □

Consider now the finite transition system $T^B_{\mathcal{A}}$ defined as the transition system $T_{\mathcal{A}}$ restricted to configurations $((s_p)_{p \in \mathcal{P}}, (w_{p,q})_{(p,q) \in \mathrm{Ch}})$ where $|w_{p,q}| \le B$ for any $(p,q) \in \mathrm{Ch}$. Since this transition system is finite, its language $L(T^B_{\mathcal{A}})$ is regular. If $\mathcal{A}$ is deadlock-free, then $\mathrm{Pref}_{B+1}(\mathcal{A}) = L(T^{B+1}_{\mathcal{A}})$, where all states in $T^{B+1}_{\mathcal{A}}$ are final. Together with Proposition 4.3, this provides us with an algorithm to test whether the CFM $\mathcal{A}$ is universally B-bounded:

Proposition 4.4. The question whether a deadlock-free CFM is universally B-bounded is a PSPACE-complete problem, provided that B is given in unary.

Proof:
First, let us note that $T^{B+1}_{\mathcal{A}}$ has an exponential number of states. Second, the set of all B-bounded linearizations of prefix MSCs is the language of a *deterministic* automaton with an exponential number of states, hence its complement also has an exponential number of states. We can decide the emptiness of the intersection of two finite automata in logarithmic space, hence we get a PSPACE algorithm for the question whether every linearization in $\mathrm{Pref}_{B+1}(\mathcal{A})$ is B-bounded.

For the lower bound, it suffices to notice that a Turing machine TM never uses more than B space iff the CFM $\mathcal{A}_{\mathrm{TM}}$ constructed before the proof of Proposition 4.1 is universally B-bounded. Since the problem of deciding whether a Turing machine is B-space bounded (with B given in unary) is PSPACE-hard, the PSPACE-hardness of our problem follows. □

5. The behavior of existentially bounded CFMs

An extension of the trace technique of Section 3 allows to obtain the equivalence between MSO and CFM within the larger setting of existentially bounded MSCs. As stated in the introduction, existentially bounded message channels circumvent the need of acknowledgments that are needed in the universally bounded case. Moreover, existential bounds are a lot more realistic when modeling one-way communication (such as e.g. in the producer-consumer setting), and the existence of such bounds amounts to the existence of some sort of fair scheduling between sends and receives, that avoids overflow of channels. On the other hand, the lack of message acknowledgments makes the proofs, in particular the CFM construction, much trickier.

The difficulty here consists in constructing a CFM that recognizes the set of all existentially B-bounded MSCs (a nondeterministic CFM accepting the set of all universally B-bounded MSCs is easily constructed, and this set can even be accepted deterministically [20, Lemma 3.14]). We do not know whether the set of existentially B-bounded MSCs can be recognized by a *deterministic* CFM. However we exhibit in this section an example that shows that deterministic, existentially B-bounded CFMs are strictly less powerful than existentially B-bounded CFMs. We end the section by a result showing that it can be decided whether a deadlock-free CFM is existentially bounded, for a given bound B.

5.1. Büchi characterization of existentially bounded behaviors

Informally, a CFM is existentially B-bounded, if the sending and receiving events can always be scheduled in such a way that the size of the channels never exceeds B. Let $\mathcal{A}$ be a CFM, and recall that $T_{\mathcal{A}}^{B}$ is the restriction of the transition system associated with $\mathcal{A}$ to runs where each configuration has channels bounded by B. By definition, $L(T_{\mathcal{A}}^{B}) \subseteq L(T_{\mathcal{A}})$.

Definition 5.1. Let B be a positive integer. An MSC M is *existentially B-bounded* if $\mathrm{Lin}_B(M) \neq \emptyset$. A set of MSCs $\mathcal{M}$ is existentially B-bounded if every $M \in \mathcal{M}$ is existentially B-bounded. A CFM $\mathcal{A}$ is called existentially B-bounded if $\mathrm{msc}(L(T_{\mathcal{A}}^{B})) = \mathrm{msc}(L(T_{\mathcal{A}}))$.

A CFM (set of MSCs, respectively) is called existentially bounded if it is existentially B-bounded for some $B > 0$.

For a set of MSCs $\mathcal{M}$, we call $X \subseteq \mathrm{Lin}(\mathcal{M})$ a set of *representative* linearizations for $\mathcal{M}$if for each $M \in \mathcal{M}$, we have $X \cap \mathrm{Lin}(M) \neq \emptyset$. In particular, if a CFM $\mathcal{A}$ is existentially B-bounded, then $L(T_{\mathcal{A}}^{B})$ is a set of representative linearizations of $\mathcal{A}$. Notice that if there exists a regular set of representatives of $\mathcal{M}$, then $\mathcal{M}$ is existentially bounded.

We start first with a characterization of existentially B-bounded MSCs. With an MSC $M = (E, \leq, \lambda)$ we asociate the binary relation on events $\prec_B \subseteq E \times E$ [21] given by $\prec_B = <_m \cup \bigcup_{p \in \mathcal{P}} <_p \cup \mathrm{rev}$, where rev is given by

$$\begin{aligned}(r, s') \in \mathrm{rev} \quad \text{iff} \qquad & s <_m r,\ \lambda(s) = \lambda(s'),\ \text{ and} \\ & |\{x \in E \mid s <_p x \leq_p s', \lambda(s) = \lambda(x)\}| = B.\end{aligned}$$

That is, the relation rev maps a receive r with $s <_m r$ to the send s' that is the B-th event with $\lambda(s') = \lambda(s)$ and $s < s'$ (if such an event exists). Hence, if M is universally B-bounded, then $\mathrm{rev} \subseteq \leq$, i.e., $\prec_B^* = \leq$. Recall the encoding defined in Section 3, that numbers the events of an MSC $M = (E, \leq, \lambda)$ modulo B, via the labeling $\lambda' : E \to \Omega$. Extending the definition from the case of universally B-bounded MSCs we denote by $\mathrm{tr}(M)$ the structure $(E, \prec_B^*, \lambda')$.

Lemma 5.1. [21] Let B be a positive integer, and $M = (E, \leq, \lambda)$ an MSC. Then M is existentially B-bounded iff the relation $\prec_B$ is acyclic. In this case, the structure $\mathrm{tr}(M)$ is a trace over (Ω, I).

Figure 2 depicts the result of applying the encoding used in Section 3 to an existentially 2-bounded MSC M. Note that in addition to the edges of the partial order we have an edge from the first occurrence of $(q?p, 0)$ to the second occurrence of $(p!q, 0)$, this edge is a rev-edge. Since M is existentially 2-bounded, the relation $\prec_2$ is acyclic by Lemma 5.1 and $\mathrm{tr}(M) = (E, \prec_2^*, \lambda)$ is precisely the trace represented in Figure 2. The reader can also easily check that $\prec_1$ is not acyclic. Notice also that $M \neq \mathrm{proj}(\mathrm{tr}(M))$, unlike the universal bounded case. Here, the trace $\mathrm{tr}(M)$ orders more events than the MSC M.

The lemma below is similar to the case of the universal B-bound.

Lemma 5.2. Let $\mathcal{A}$ be an existentially B-bounded CFM. Then there exists a regular language of traces L over (Ω, I) such that $\mathcal{L}(\mathcal{A}) = \mathrm{msc}(\mathrm{proj}(\mathrm{Lin}(L)))$.

The next theorem provides the characterization of existentially bounded CFMs (with given channel bound) in terms of monadic second-order logic and of regular linearizations. For lack of space, we have omitted again the third characterization, in terms of globally-cooperative CMSC-graphs [16].

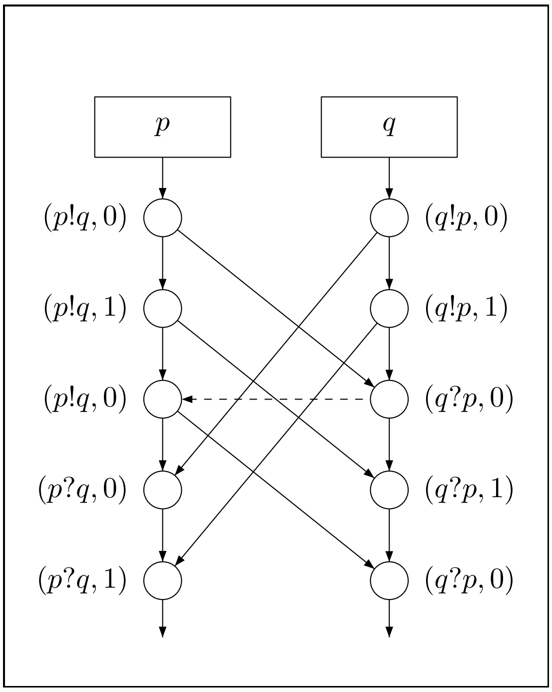

Figure 2. Trace $\mathrm{tr}(M)$ associated with an existentially 2-bounded MSC.

The results given below were obtained in [14]. Again, the most difficult part of the proof is the construction of a CFM from a regular set $\mathrm{Lin}_B(\mathcal{M})$. The proof uses the trace language from Lemma 5.2, but an additional difficulty arises by the fact that the partial order of the MSC M is weaker than the partial order of its trace structure $\mathrm{tr}(M)$.

Theorem 5.1. [14] Let B be a positive integer and $\mathcal{M}$ a set of existentially B-bounded MSCs. Then the following assertions are equivalent:

1. $\mathrm{Lin}_B(\mathcal{M})$ is regular.
2. $\mathcal{M}$ is the language of some CFM.
3. $\mathcal{M}$ is the language of some MSO($\leq$) formula.
4. $\mathcal{M}$ is the language of some formula of EMSO($(\lessdot_p)_{p\in\mathcal{P}}, <_m$), EMSO($\leq$), or EMSO($\leq, <_m$), respectively.

The proof of the theorem above follows the main lines of the universally bounded case. As previously, the main difficulty is the construction of the CFM from the representative set $\mathrm{Lin}_B(\mathcal{M})$. Once again, the idea is to apply first Zielonka's construction of asynchronous automata to the trace language obtained by Lemma 5.2. In addition, we need to solve two more problems: first, the simulation of the asynchronous automaton by the CFM is non-deterministic, since the information conveyed by the rev-edges in the runs of the asynchronous automaton has to be guessed by the receiver (recall that these edges do not exist in the MSC). Second, a CFM recognizing all existentially B-bounded MSCs must be constructed. Both parts involve non-deterministic guesses in the CFM, and the example in the next section shows that non-determinism is unavoidable. The equivalence between the different EMSO-logics can be shown as in the universally bounded case (see end of Section 4.1 or [14, Prop. 6.2]). The only difference is the formula that expresses that the model is an existentially B-bounded MSC. An MSO-formula for the set of all existentially B-bounded MSCs is easily build (it expresses that $\prec_B$ has to be acyclic). The EMSO formula is more involved, it uses the CFM that accepts all existentially B-bounded MSCs.

5.2. Deterministic CFMs are strictly weaker

Let $\mathcal{P} = \{0, 1, 2, 3, 4\}$. All MSCs we will consider in the following send only messages from process 0 to processes 1 and 2, from process 1 to 3, and from process 2 to 4. For $p \in \{0, \ldots, 4\}$ let $\pi_p(M)$ denote the projection of M onto the events of process p.

Let $\mathcal{L}_0$ consist of all MSCs such that

- $\pi_0(M) \in [(0!1)(0!2)]^*$,
- $\pi_1(M) \in [(1?0)((1!3) + (1!3)(1!3))]^*$, and $\pi_2(M) \in [(2?0)((2!4) + (2!4)(2!4))]^*$,
- $\pi_3(M) \in (3?1)^*$ and $\pi_4(M) \in (4?2)^*$.

Thus, process 0 will send alternately to 1 and 2. Process 1 will perform one or two send actions $1!3$ between any two receive actions $1?0$ and similarly for process 2. Finally, processes 3 and 4 will just receive messages from 1 and 2, respectively.

Now define the mapping $\phi : \Sigma^* \to \Sigma^*$ by renaming 2 into 1 and 4 into 3. Let $\mathcal{L} \subseteq \mathcal{L}_0$ consist of all those MSCs from $\mathcal{L}_0$ where the sequence of actions of processes 1 and 2 are the same modulo ϕ, i.e., $\pi_1(M) = \phi(\pi_2(M))$.

Proposition 5.1. The MSC language $\mathcal{L}$ can be accepted by some CFM, but not by any deterministic CFM.

Proof:
A CFM for $\mathcal{L}$ is easily defined, by letting process 0 decide whether process 1 and 3 send one or two messages each. Process 0 sends non-deterministically either the message ”1”, or the message ”2” to processes 1 and 2 each. On receiving message ”i”, process 1 sends precisely i messages to process 3 (and similarly for processes 2 and 4).

Now suppose that $\mathcal{A}$ is a deterministic CFM that accepts $\mathcal{L}$. Then there are distinct MSCs M_1 and M_2 from $\mathcal{L}$ such that

- $\pi_0(M_1) = \pi_0(M_2)$ and
- $\mathcal{A}$ terminates in the same accepting global state when executing M_1 and M_2.

Because of the first of these requirements, there exists an MSC M such that

1. $\pi_0(M) = \pi_0(M_1) = \pi_0(M_2)$, $\pi_1(M) = \pi_1(M_1)$, $\pi_3(M) = \pi_3(M_1)$, and
2. $\pi_2(M) = \pi_2(M_2)$, $\pi_4(M) = \pi_4(M_2)$.

Let ρ_1 and ρ_2 be the unique (and successful) runs of $\mathcal{A}$ on M_1 and M_2, respectively. Recall that $\mathcal{A}$ is deterministic and process 0 does not perform any receive events in M_1 or in M_2. Hence ρ_1 and ρ_2 behave the same on process 0. Hence we can construct a run ρ of $\mathcal{A}$ on M as follows:

1. on processes 0, 1, and 3, it behaves like ρ_1 and

2. on processes 0, 2, and 4, it behaves like ρ_2.

Because of the second of the above requirements, the run ρ is successful, i.e., M is accepted by $\mathcal{A}$ and therefore an element of $\mathcal{L}$.

Since M_1 and M_2 are distinct with $\pi_0(M_1) = \pi_0(M_2)$, we have either $\pi_1(M_1) \neq \pi_1(M_2)$ or $\pi_2(M_1) \neq \pi_2(M_2)$. We consider the case $\pi_1(M_1) \neq \pi_1(M_2)$ in more detail, the other case is dealt with similarly. Since the only actions performed by process 1 are $1?0$ and $1!3$, we obtain $\pi_1(M) = \pi_1(M_1) \neq \pi_1(M_2)$, but $\pi_2(M) = \pi_2(M_2)$, hence $\pi_1(M) \neq \phi(\pi_2(M))$, which contradicts $M \in \mathcal{L}$. □

Theorem 5.2. Non-deterministic existentially bounded CFMs are strictly more expressive than deterministic existentially bounded CFMs.

5.3. Testing existential bounds

In this section we consider the test whether a given CFM is existentially bounded. We show that the decidability and complexity of deciding universal and existential channel bounds is the same, albeit the fact that proofs are more involved in the existential case.

The proof of Proposition 4.1 yields quickly a similar result for the existentially bounded case:

Proposition 5.2. Let $B > 0$. It is undecidable whether a deterministic CFM is existentially B-bounded.

Proposition 5.3. It is undecidable whether a deterministic and deadlock-free CFM is existentially bounded.

Proof:
It suffices to reconsider the proof of Proposition 4.2, and to notice that $\mathcal{A}_{\text{TM}}$ is actually existentially bounded iff the Turing machine TM has a bound on the size of its reachable configurations. □

We consider now the question whether a deadlock-free CFM is existentially B-bounded, for given B. We already know from Proposition 3.1 that an MSC M is existentially B-bounded iff the optimal linearization $\text{OPT}(M)$ is B-bounded. We would like to mimic the proof of Proposition 4.4, that showed how to test (in polynomial space) whether a deadlock-free CFM is universally B-bounded. Notice however that $\text{Pref}(\mathcal{A})$ is not the right set to deal with, since the property of being existentially B-bounded is not inherited by prefixes. One can observe this phenomenon on an MSC with two processes $1, 2$, where process 1 starts by sending $B + 1$ consecutive messages to 2. The prefix MSC consisting of the $B + 1$ sends has of course no B-bounded linearization.

Let M be an MSC and consider a prefix $N = (E, \leq, \lambda)$ of M. We define N_c as the restriction of N to the set of matched events $E \setminus \{e \in E \mid \forall f \in E : \ e \not<_m f \wedge f \not<_m e\}$ of N (N_c contains all receives of N since N is a prefix). The set $\text{CPref}(M)$ consists of all MSCs N_c, associated with prefixes N of M. Alternatively, we can construct $\text{CPref}(M)$ incrementally: For MSCs $M = (E, \leq, \lambda)$ and N we write $M \rightarrow N$ if there exists some maximal event $r \in E$ such that N is the restriction of M to $E \setminus \{s, r\}$ where $s \in E$ is the unique event with $s <_m r$ (i.e., N is obtained from M by deleting some message with maximal receive). Note that neither N_c nor N need to be prefixes of M.

Lemma 5.3. $\mathrm{CPref}(M)$ is the least set of MSCs that contains M and, with $N_1 \to N_2$ and $N_1 \in \mathrm{CPref}(M)$ also contains N_2.

Proof:
It is easy to see that $\mathrm{CPref}(M)$ is closed under $\to$, which gives one inclusion. For the converse, let $N = (E, \leq, \lambda)$ be a prefix of M. Then N_c can be obtained from M by iterating $\to$ until all maximal receive events belong to E. $\Box$

For a set of MSCs $\mathcal{M}$ we write $\mathrm{CPref}(\mathcal{M})$ for $\bigcup_{M \in \mathcal{M}} \mathrm{CPref}(M)$, and for a CFM $\mathcal{A}$ we write $\mathrm{CPref}(\mathcal{A})$ instead of $\mathrm{CPref}(\mathcal{L}(\mathcal{A}))$. Finally, for $B > 0$ we denote by $\mathrm{CPref}_B(\mathcal{A})$ the subset of existentially B-bounded MSCs in $\mathrm{CPref}(\mathcal{A})$.

Proposition 5.4. For any MSC M we have:

- If M is existentially B-bounded, then every $N \in \mathrm{CPref}(M)$ is existentially B-bounded.
- If M is not existentially B-bounded, then there exists some $N \in \mathrm{CPref}(M)$ that is existentially $(B+1)$-bounded, but not existentially B-bounded.

Proof:
Let w be a B-bounded linearization of M. Deleting in w all symbols that do not occur in N yields a linearization of N which is B-bounded.

For the second statement, suppose that $M = (E, \leq, \lambda)$ is not existentially B-bounded. We reason by induction on the size of M. Consider two events s, r of M that form a message, i.e., $s <_m r$, and such that r is maximal in M. Then let $M' = M \setminus \{s, r\}$ be the restriction of M to the events in $E \setminus \{s, r\}$. If M' is existentially B-bounded, then M is existentially $(B+1)$-bounded; in this case we set $N = M$. Else, by induction we obtain some $N' \in \mathrm{CPref}(M')$ that is existentially $(B+1)$-bounded, but not existentially B-bounded. With Lemma 5.3 we obtain $\mathrm{CPref}(M') \subseteq \mathrm{CPref}(M)$, hence N' is the desired result. $\Box$

Corollary 5.1. Let $\mathcal{A}$ be a CFM, and $B > 0$. Then $\mathcal{A}$ is existentially B-bounded if and only if every MSC in $\mathrm{CPref}_{B+1}(\mathcal{A})$ is existentially B-bounded.

Proof:
If $\mathcal{A}$ is existentially B-bounded, then so is $\mathrm{CPref}(\mathcal{A})$ by Proposition 5.4. Therefore we have $\mathrm{CPref}(\mathcal{A}) = \mathrm{CPref}_B(\mathcal{A}) = \mathrm{CPref}_{B+1}(\mathcal{A})$. Conversely, if $\mathcal{A}$ is not existentially B-bounded we obtain using Proposition 5.4 some $N \in \mathrm{CPref}_{B+1}(\mathcal{A})$ that is not existentially B-bounded. $\Box$

Our next (intermediate) aim is to show that, provided the CFM $\mathcal{A}$ is deadlock-free, the set of $(B+1)$-bounded linearizations of $\mathrm{CPref}(\mathcal{A})$ is regular and can be accepted by an automaton with exponentially many states.

To this aim, we first construct an infinite transition system with ε-transitions $T'(\mathcal{A})$ for the set of all linearizations of $\mathrm{CPref}(\mathcal{A})$. The idea is to add a flag for each channel. If this flag is raised, any sends to this channel are ignored (i.e., they give rise to ε-transitions). Otherwise, $T'(\mathcal{A})$ works as the usual transition system $T_{\mathcal{A}}$ associated with $\mathcal{A}$.

The states of $T'(\mathcal{A})$ are of the form $S = ((s_p)_{p \in \mathcal{P}}, (w_{p,q})_{(p,q) \in \mathrm{Ch}}, (f_{p,q})_{(p,q) \in \mathrm{Ch}})$, where s_p is a local state of $\mathcal{A}_p$, $w_{p,q} \in C^*$ is a channel content, and the last component $f_{p,q}$ is a flag for channel (p,q), taking values 0 or 1. The state S is initial if $s_p = \iota_p$ is locally initial, $w_{p,q}$ is empty, and $f_{p,q}$ is arbitrary; S is accepting if all channels are empty. There are three types of transitions: send, receive, and ε-transitions. To define these transitions, let $S = ((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}}, (f_{p,q})_{(p,q)\in\mathrm{Ch}})$ and $S' = ((s'_p)_{p\in\mathcal{P}}, (w'_{p,q})_{(p,q)\in\mathrm{Ch}}, (f'_{p,q})_{(p,q)\in\mathrm{Ch}})$ be states of $T'(\mathcal{A})$. We have a transition from S to S' provided that $f_{r,s} \leq f'_{r,s}$ for all channels $(r,s) \in \mathrm{Ch}$ and one of the following holds:

1. $S \xrightarrow{a} S'$ is a *receive transition* whenever $a = q?p$ and $((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}}) \xrightarrow{a,c} ((s'_p)_{p\in\mathcal{P}}, (w'_{p,q})_{(p,q)\in\mathrm{Ch}})$ is a transition of $T_{\mathcal{A}}$, for some control message $c \in C$.

2. $S \xrightarrow{a} S'$ is a *send transition* whenever $a = p!q$, $f_{p,q} = 0$, and $((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}}) \xrightarrow{a,c} ((s'_p)_{p\in\mathcal{P}}, (w'_{p,q})_{(p,q)\in\mathrm{Ch}})$ is a transition of $T_{\mathcal{A}}$, for some control message $c \in C$.

3. $S \xrightarrow{\varepsilon} S'$ is an *ε-transition* whenever there exists a channel $(p,q) \in \mathrm{Ch}$ with $f_{p,q} = 1$ and a transition $((s_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}}) \xrightarrow{p!q,c} ((s'_p)_{p\in\mathcal{P}}, (w''_{p,q})_{(p,q)\in\mathrm{Ch}})$ of $T_{\mathcal{A}}$, for some control message $c \in C$ and channel content $w''_{r,s}$. Moreover, $w_{r,s} = w'_{r,s}$ for all channels $(r,s) \in \mathrm{Ch}$.

Thus, receives of $T_{\mathcal{A}}$ are simulated by $T'(\mathcal{A})$ without any change. Send actions however, can be transformed into ε-transitions, provided that the flag is set. At any moment, the flag can be raised for any channel.

Lemma 5.4. If the CFM $\mathcal{A}$ is deadlock-free, then $\mathrm{Lin}(\mathrm{CPref}(\mathcal{A})) = L(T'(\mathcal{A}))$, i.e., the transition system $T'(\mathcal{A})$ accepts precisely the linearizations of elements of $\mathrm{CPref}(\mathcal{A})$.

Proof:
Let $N \in \mathrm{CPref}(\mathcal{A})$ and $w \in \mathrm{Lin}(N)$. Then there exists $uv \in \mathrm{Lin}(\mathcal{L}(\mathcal{A})) = L(T_{\mathcal{A}})$ such that w results from u by deleting all sends that are not matched in u. Consider a path in $T_{\mathcal{A}}$ that corresponds to uv. The prefix of this path corresponding to u gives rise to a w-labeled path in $T'(\mathcal{A})$ (transitions that correspond to unmatched sends get replaced by ε-transitions). This path in $T'(\mathcal{A})$ ends in a state with empty channels, i.e., it is accepting. Hence $\mathrm{Lin}(\mathrm{CPref}(\mathcal{A})) \subseteq L(T'(\mathcal{A}))$.

For the other implication, consider some accepting path in $T'(\mathcal{A})$ for w, starting in the state $((\iota_p)_{p\in\mathcal{P}}, (\varepsilon)_{(p,q)\in\mathrm{Ch}}, (f_{p,q})_{(p,q)\in\mathrm{Ch}})$ and leading to $((s'_p)_{p\in\mathcal{P}}, (\varepsilon)_{(p,q)\in\mathrm{Ch}}, (f'_{p,q})_{(p,q)\in\mathrm{Ch}})$. Note that this path contains some ε-transitions on channels whose flag is set at some point. These ε-transitions correspond to “hidden sends”. Let $u \in (\Sigma \times C)^*$ be obtained from w by adding all these hidden sends at the appropriate positions, and adding the control messages used by the accepting path in $T'(\mathcal{A})$. Then, in the transition system $T_{\mathcal{A}}$, there is a path from $((\iota_p)_{p\in\mathcal{P}}, (\varepsilon)_{(p,q)\in\mathrm{Ch}})$ to $S = ((s'_p)_{p\in\mathcal{P}}, (w_{p,q})_{(p,q)\in\mathrm{Ch}})$ for some channel contents $w_{p,q}$, labeled by u. Since the CFM $\mathcal{A}$ is deadlock-free, there exists also a path in $T_{\mathcal{A}}$ from S to some accepting state, labeled by $v \in (\Sigma \times C)^*$. Thus, uv labels an accepting path of $T_{\mathcal{A}}$, hence the MSC associated with uv is in $\mathcal{L}(\mathcal{A})$. Since w is obtained from the Σ-projection of u by deleting all unmatched sends, this proves $w \in \mathrm{Lin}(\mathrm{CPref}(\mathcal{A}))$ and therefore $N = \mathrm{msc}(w) \in \mathrm{CPref}(\mathcal{A})$. Hence we proved $L(T'(\mathcal{A})) \subseteq \mathrm{Lin}(\mathrm{CPref}(\mathcal{A}))$ and therefore the equality of these two sets. □

Proposition 5.5. The question whether a deadlock-free CFM is existentially B-bounded is a PSPACE-complete problem, provided that B is given in unary.

Proof:
By Cor. 5.1 and Prop. 3.1, we have to check that any word $\mathrm{OPT}(M)$ from $\mathrm{Lin}_{B+1}(\mathrm{CPref}(\mathcal{A}))$ is B-bounded.

Restricting the transition system $T'(\mathcal{A})$ to those states whose channels contain at most $B + 1$ messages, we obtain a finite automaton $T_{B+1}(\mathcal{A})$ with exponentially many states that accepts

$\mathrm{Lin}_{B+1}(\mathrm{CPref}(\mathcal{A}))$. Using Proposition 3.2, we can construct an automaton $\mathcal{B}$ with exponentially many states that accepts the intersection of $\mathrm{Lin}_{B+1}(\mathrm{CPref}(\mathcal{A}))$ with the set of optimal linearizations $\{\mathrm{OPT}(M) \mid M \text{ MSC}\}$ (it suffices to complement the automaton of Proposition 3.2). Note that there exists a *deterministic* automaton $\mathcal{C}$ with exponentially many states that accepts the set of B-bounded words. Hence we can test whether $L(\mathcal{B}) \subseteq L(\mathcal{C})$ in polynomial space.

For the lower bound, we apply a similar argument as in the proof of Proposition 4.4. □

6. Conclusion

It follows from Theorem 5.1 that CFMs can be complemented relative to the set of existentially B-bounded MSCs, for any bound B. We do not know how to prove this explicitly without exploiting the equivalence to MSO, which is trivially closed under negation. Another consequence of Theorem 5.1 is that several interesting model checking instances are decidable. We can check 1) whether all existentially B-bounded behaviors of a CFM satisfy an MSO formula, for any bound B, and 2) whether a regular set of B-bounded linearizations is included in (intersects, respectively) the language of a CFM.

Figure 3 summarizes the results obtained for the problem of testing channel bounds (with and without an explicitly provided bound B, respectively). Note that the undecidability results hold even for deterministic CFMs.

	$\forall$ B-bound	$\exists$ B-bound	$\forall$-bound	$\exists$-bound
Arbitrary CFM	undecidable	undecidable	undecidable	undecidable
Deadlock-free CFM	PSPACE	PSPACE	undecidable	undecidable

Figure 3. Testing boundedness

Acknowledgement: We thank the referees for several pertinent and useful remarks.

References

[1] P. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[2] P. Abdulla and B. Jonsson. Undecidable Verification Problems for Programs with Unreliable Channels. *Information and Computation*, 130(1):71–90, 1996.

[3] R. Alur and M. Yannakakis. Model checking of message sequence charts. In *CONCUR'99*, LNCS 1664, pp. 114-129, 1999.

[4] C. Baier, N. Bertrand and Ph. Schnoebelen. Symbolic verification of communicating systems with probabilistic message losses: liveness and fairness. In *FORTE'06*, LNCS 4229, pp. 212-227, 2006.

[5] A. Bouajjani and P. Habermehl. Symbolic Reachability Analysis of FIFO-Channel Systems with Nonregular Sets of Configurations. *Theoretical Computer Science*, 221(1-2):211–250, 1999.

[6] B. Boigelot, P. Godefroid, B. Willems and P. Wolper. The Power of QDDs. In *SAS'97*, LNCS 1302, pp. 172-186, 1997.

[7] B. Boigelot and P. Godefroid. Symbolic Verification of Communication Protocols with Infinite State Spaces using QDDs. *Formal Methods in System Design*, 14(3):237-255 (1999).

[8] B. Bollig and M. Leucker. Message-Passing Automata are expressively equivalent to EMSO Logic. *Theoretical Computer Science* 358(2-3):150-172 (2006). (An extended abstract appeared under the same title in *CONCUR'04*, LNCS 3170, pp. 146-160, 2004.)

[9] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323-342, 1983.

[10] G. Cécé and A. Finkel. Programs with quasi-stable channels are effectively recognizable. In *CAV'97*, LNCS 1254, pp. 304–315, 1997.

[11] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[12] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.

[13] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1,2):63–92, 2001.

[14] B. Genest, D. Kuske, and A. Muscholl. A Kleene Theorem and Model Checking for a Class of Communicating Automata. *Information and Computation*, 204(6):920-956, 2006.

[15] E. Gunter, A. Muscholl, and D. Peled. Compositional Message Sequence Charts. *International Journal on Software Tools for Technology Transfer (STTT)* 5(1): 78-89 (2003). (An extended abstract appeared in *TACAS'01*, LNCS 2031, pp. 496–511, 2001.)

[16] B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state High-level MSCs: Model checking and realizability. *J. Comput. Syst. Sci.* , 72(4): 617-647, 2006.

[17] J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P. Thiagarajan. A Theory of Regular MSC Languages. *Information and Computation*, 202(1):1–38, 2005.

[18] ITU-TS recommendation Z.120, Message Sequence Charts, Geneva, 1999.

[19] S.R. Kosaraju. Decidability of reachability in vector addition systems. In *STOC'84*, pp. 267-281. 1984.

[20] D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187:80–109, 2003.

[21] M. Lohrey and A. Muscholl. Bounded MSC communication. *Information and Computation*, 189:135–263, 2004.

[22] S. Leue, R. Mayr, and W. Wei. A scalable incomplete test for the boundedness of UML RT models. In *TACAS'04*, LNCS 2988, pp. 327–341, 2004.

[23] R. Morin. Recognizable Sets of Message Sequence Charts. In *STACS'02*, LNCS 2285, pp. 523-534, 2002.

[24] P. Madhusudan. Reasoning about Sequential and Branching Behaviours of Message Sequence Graphs. In *ICALP'01*, LNCS 2076, pp. 809-820, 2001.

[25] P. Madhusudan and B. Meenakshi. Beyond Message Sequence Graphs In *FSTTCS'01*, LNCS 2245, pp. 256-267, 2001.

[26] E. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. of Comp.*, 13:441-459 (1984).

[27] A. Mazurkiewicz. Concurrent program schemes and their interpretation. Technical report, DAIMI Report PB-78, Aarhus University, 1977.

[28] A. Muscholl and D. Peled. Message sequence graphs and decision problems on Mazurkiewicz traces. In *MFCS'99*, LNCS 1672, pp. 81-91, 1999.

[29] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letter*, 83(5):251–261, 2002.

[30] Ph. Schnoebelen. The Verification of Probabilistic Lossy Channel Systems. In Validation of Stochastic Systems: A Guide to Current Research, LNCS 2925, pp. 445-465, 2004.

[31] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proceedings of a workshop of the ESPRIT BRA No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS) 1989*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990.

[32] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. – Informatique Théorique et Applications*, 21:99-135, 1987.